Partial Local FriendQ Multiagent Learning: Application to Team Automobile Coordination Problem

Julien Laumonier and Brahim Chaib-draa

DAMAS Laboratory, Department of Computer Science
and Software Engineering, Laval University, Canada
{jlaumoni;chaib}@damas.ift.ulaval.ca

Abstract. Real world multiagent coordination problems are important issues for reinforcement learning techniques. In general, these problems are partially observable and this characteristic makes the solution computation intractable. Most of the existing approaches calculate exact or approximate solutions using the world model for only one agent. To handle a special case of partial observability, this article presents an approach to approximate the policy measuring a degree of observability for pure cooperative vehicle coordination problem. We compare empirically the performance of the learned policy for totally observable problems and performances of policies for different degrees of observability. If each degree of observability is associated with communication costs, multiagent system designers are able to choose a compromise between the performance of the policy and the cost to obtain the associated degree of observability of the problem. Finally, we show how the available space, surrounding an agent, influence the required degree of observability for near-optimal solution.

1 Introduction

In real world cooperative multiagent problem, each agent has often a partial view of the environment. If communication is possible without cost, the multiagent problem becomes totally observable and can be solved optimally using reinforcement learning techniques. However, if the communication has a cost, the multiagent system designer has to find a compromise between increasing the observability and consequently the performance of the learned policy and the total cost of the multiagent system. Some works present formal models to take into account the communication decision into the multiagent decision problem [1], [2]. For the non-cooperative multiagent problem, some works introduce also explicitly communication into general sum games [3] [4].

To allow the multiagent system designer to choose a compromise between performance and partial observability, we propose, in this article, to take into account the degree of observability for a cooperative multiagent system by measuring the performance of the associated learned policy. In this article, the degree

of observability is defined as the agent's vision distance. Obviously, decreasing the observability reduces the number of accessible states for agents and therefore decrease the performance of the policy. A subclass of coordination problems is purely cooperative multiagent problems where all agents have the same utility function. This kind of problems is known as team games [5]. In this kind of games, if we consider problems where agents' designer neither has the transition function nor the reward function, we can use learning algorithms. Many of these algorithms have been proven to converge to Pareto-optimal equilibrium such as Friend Q-learning [6] and OAL [7]. Consequently, one can take an optimal algorithm to find the policy for the observable problem.

As we restrict our problem to team problems, the following assumptions are defined: (1) Mutually exclusive observations, each agent sees a partial view of the real state but all agents together see the real state. (2) Possible communication between agents but not considered as an explicit part of the decision making. (3) The problem involves only negative interactions between agents. One problem which meets these assumptions is the choosing lane decision problem [8] related to Intelligent Transportation Systems [9]. In this problem, some vehicles, which have to share a part of the road, decide to change lane or not, in order to increase traffic flow and reduce collisions. In this article, we show empirically that the performance of the learning algorithm is closely related to the degree of observability. Moreover, we show that there exists a relation between the available space for each agent and a "correct" degree of observability that allow a good policy approximation.

This paper is organized as follows. Section 2 describes the formal model and algorithms used in our approach. Section 3 describes the vehicle coordination problem with more details. Section 4 explains our approach by introducing a partial local state. Section 5 provides the results and a discussion about them. Section 6 presents the related works and Section 7 concludes.

2 Formal Model and Algorithms

Reinforcement learning allows an agent to learn by interacting with its environment. For a mono agent system, the basic formal model for reinforcement learning is Markov Decision Process [10]. Using this model, Q-Learning algorithm calculates the optimal values of the expected reward for the agent in a state s if the action a is executed. To do this, the following update function is used:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha[r + \gamma \max_{a \in A} Q(s',a)]$$

where r is the immediate reward, s' is the next state and α is the learning rate. An *episode* is defined as a sub-sequence of interaction between the agent and its environment.

On the other hand, game theory studies formally the interaction of rational agents. In a one-stage game, each agent i has to choose an action to maximize its own utility $U^i(a^i, a^{-i})$ which depends on the others' actions a^{-i}. An action can

be *mixed* if the agent chooses it with a given probability and can be *pure* if it is chosen with probability 1. In game theory, the solution concept is the notion of equilibrium. For an agent, the equilibria are mainly based on the best response to other's actions. Formally, an action a^i_{br} is a best response to actions a^{-i} of the others agents if

$$U^i(a^i_{br}, a^{-i}) \geq U^i(a'^i, a^{-i}),\ \forall a'^i.$$

The set of best responses to a^{-i} is noted $BR^i(a^{-i})$.

The Nash equilibrium is the best response for all agents. Formally, a joint action a_N, which regroups the actions for all agents, is a Nash equilibrium if

$$\forall i,\ a^i_N \in BR^i(a^{-i})$$

where a^i_N is the action of the i^{th} agent in the Nash equilibrium and a^{-i}_N is the actions of other agents at Nash equilibrium. A solution is Pareto optimal if there does not exist any other solution such that one agent can improve its reward without decreasing the reward of another.

The model which combines reinforcement learning and game theory, is *stochastic games* [11]. This model is a tuple $< Ag, S, A^i, \mathcal{P}, \mathcal{R}^i >$ where

- Ag is the set of agents where $\text{card}(Ag) = N$,
- $S = \{s_0, \cdots, s_M\}$ is the finite set of states where $|S| = M$,
- $A^i = \{a^i_0, \cdots, a^i_p\}$ is the finite set of actions for the agent i,
- $\mathcal{P} : S \times A^1 \times \cdots \times A^N \times S \rightarrow \Delta(S)$ is the transition function from current state, agents actions and new state to probability distribution over state,
- $\mathcal{R}^i : S \times A^1 \times \cdots \times A^N \rightarrow \mathbb{R}$ is the immediate reward function of agent i. In team Markov games, $\mathcal{R}^i = \mathcal{R}$ for all agents i.

Among the algorithms which calculate a policy for team Markov games, Friend Q-Learning algorithm, introduced by Littman [6], allows to build a policy which is a Nash Pareto optimal equilibrium in team games. More specifically, this algorithm, based on Q-Learning, uses the following function for updating the Q-values:

$$Q(s, \boldsymbol{a}) = (1 - \alpha)Q(s, \boldsymbol{a}) + \alpha[r + \gamma \max_{\boldsymbol{a} \in \boldsymbol{A}} Q(s', \boldsymbol{a})]$$

with $\boldsymbol{a}$, the joint action for all agents ($\boldsymbol{a} = (a^1, \cdots, a^N)$).

3 Problem Description

Vehicle coordination is a sub-problem of Intelligent Transportation Systems which aims to reduce congestion, pollution, stress and increase safety of the traffic. Coordination of vehicles is a real world problem with all the difficulties that can be encountered: partially observable, multi-criteria, complex dynamic, and continuous. Consequently, we establish many assumptions to apply the multiagent reinforcement learning algorithm to this problem.

The vehicle coordination problem presented here is adapted from Moriarty and Langley [8]. More precisely, three vehicles, each of them represented by an agent, have to coordinate to maintain velocity and to avoid collisions. Each vehicle is represented by a position and a velocity and can change lane to the left, to the right or stay on the same lane. The objective for a learning algorithm is to find the best policy for each agent in order to maximize the common reward which is the average velocity at each turn and to avoid collision.

Figure 1 represents the initial state. The dynamic, the state and the actions are sampled in the easiest way. The vehicles' dynamic are simplified to the following first order equation with only velocity $y(t) = v \times t + y_0$. For this example, we simulate the road as a ring meaning that a vehicle is placed on the left side when it quits through the right side. The state of the environment is described by the position x^i, y^i and the velocity v^i of each agent i. Collisions occur when two agents are in the same tile. The agents do not know the transitions between states which is calculated according to the velocities of the agents and their actions. At every step, each vehicle tries to accelerate until a maximum of 5 m/s is reached. If another vehicle is in front of him, the agent in charge of the vehicle, sets its velocity to the front vehicle's velocity. At each step, a vehicle can choose three actions: stay on the same lane, change to the right lane and change to the left lane. Each episode has a maximum of 10 steps. The reward at each step is set to the average velocity among all vehicles. If a collision occurs, the episode stops. The size of the set of states is in $O((X \times Y \times |V|)^N)$ with X the number of lane, Y the length of the road, V the set of possible velocity and N the number of agents. We assume, in this problem, that each agent is able to see only its own local state (position, velocity). To obtain the states of other agents, we assume that communication is needed.

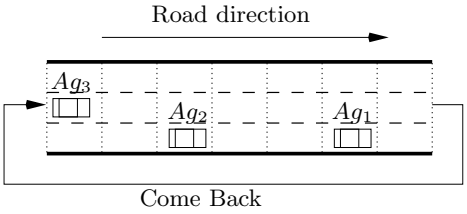

Fig. 1. Initial state for problem

4 Partial Observability

In this section, we introduce our approach describing Friend Q-learning algorithm with a local view for the agents. Then, we introduce the same algorithm that use the partial local view for a distance d. This partial local view can reduce the set of state and/or the set of joint actions. If no reduction is done, the exact

algorithm associated is Friend Q-learning. When only the set of states is reduced, we propose Total Joint Actions Q-learning (TJA). From this algorithm, we reduce the set of joint actions and we propose another algorithm: Partial Joint Actions Q-learning (PJA). In this article, we do not consider the reduction of joint actions alone, because this reduction is lower than the reduction of the set of states.

4.1 FriendQ with a local view

To introduce partial observability, we use the notion of local Q-Value and local state. Each agent uses the same algorithm but on different state. A local state is defined from the real state of the multiagent system for a center agent. All other agents positions are defined relatively to this central agent. This means that the same real state belongs to the set S will give different local states. For an agent i, the set possible local state is S^i. We introduce a function f^i which transforms the real state s to a local state s^i for agent i. Formally, $\forall s \in S, \exists s^i \in S^i$ such that $f^i(s) = s^i$ for all agents i. In this version of the algorithm, each agent uses Friend Q-learning algorithm as described in section 2 but by updating its Q-values for the local states and not for the real state.

4.2 FriendQ with a partial local view

To measure the effect of partial observability on the performance we define the partial state centered on one agent by introducing a distance of observalibity d. Consequently, the initial problem becomes a d-partial problem. The distance d can be viewed as an influence area for the agent. Increasing this distance increases the degree of observability. We define d_{total} as the maximal possible distance of observability for a given problem. Moreover, from a communication point of view, in real world problems, the communication cost between two agents depends on the distance between them. Communicating with a remote agent is costlier than with a close agent.

In d-partial problem, the new state is defined as the observation of the center agent for a range d. More precisely, an agent j is in the partial state of a central agent i if its distance is lower or equal than d from the central agent i. Formally, the function f_d^i uses the parameter d to calculate the new local state. Figure 2 provides an example of the application of f_d^i on a state s and get the result partial states for each agent with a distance $d = 2$. Agent 1 sees only Agent 3 but Agent 3 sees both other agents. The new size of the set of state is $O(((2d+1)^2 \times V)^N)$. The number of state is divided by around $(Y/(2d+1))^N$, if we neglect the number of lanes which is often small compared to the length of the road.

TJA Q-Learning In a first step, as in classical Friend Q-learning, we consider an algorithm that takes into account the complete joint actions. This assumption implies that all agents are able to communicate their actions to others at each

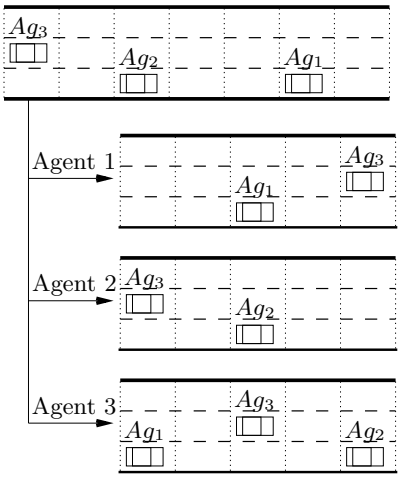

Fig. 2. State and Partial States for $d = 2$

step without cost. The Q-value update function is now :

$$Q(f_d^i(s), \boldsymbol{a}) = (1-\alpha)Q(f_d^i(s), \boldsymbol{a}) + \alpha[r + \gamma \max_{\boldsymbol{a} \in \boldsymbol{A}} Q(f_d^i(s'), \boldsymbol{a})]$$

for agent i. When $d = d_{total}$, we have a small reduction factor on the state set of XY, because we do not take into account, in our specific problem, the absolute position of the center agent.

PJA Q-learning In a second step, the algorithm takes into account only the actions where agents are in the partial local view as specified by d. This reduce dramatically the number of joint actions which have to be tested during the learning. This partial local observability allows us to consider a variable number of agents in the multiagent system.

Formally, we define a function g^i which transforms the joint action $\boldsymbol{a}$ into a partial joint action $g_d^i(\boldsymbol{a}, s)$. This partial joint action contains all actions of agent which are in the distance d of agent i. The Q-value update function is now :

$$Q(f_d^i(s), g_d^i(\boldsymbol{a}, s)) = (1-\alpha)Q(f_d^i(s), g_d^i(\boldsymbol{a}, s)) + \alpha[r + \gamma \max_{\boldsymbol{a_d} \in G_d^i(\boldsymbol{A}, S)} Q(f_d^i(s'), \boldsymbol{a_d})]$$

for agent i where $G_d^i(\boldsymbol{A}, S)$ returns the set of joint actions with a central agent i and a distance d. We can see that the result of the partial joint action depends on the current state.

5 Results

In this section, we compare empirically the performance of the totally observable problem (FriendQ) and the performance of approximated policy (TJA and PJA).

We present three kind of results: first of all, we compare the algorithms on a small problem P_1 defined by size $X = 3$, $Y = 7$, the set of velocities $V = 0, \cdots, 5$ and the number of agents $N = 3$. Consequently, in this problem, the maximal distance that we can use to approximate the total problem is $d_{total} = 3$. The 3-partial state is a local representation of the totally observable state because we are sure that all agents are visible from others in this representation. In the initial state (Figure 1), velocities of the agents are $v^1 = 1$, $v^2 = 2$ and $v^3 = 3$. We present, for all results, the average total sum reward over 25 learnings with each episode lasts 10 steps.

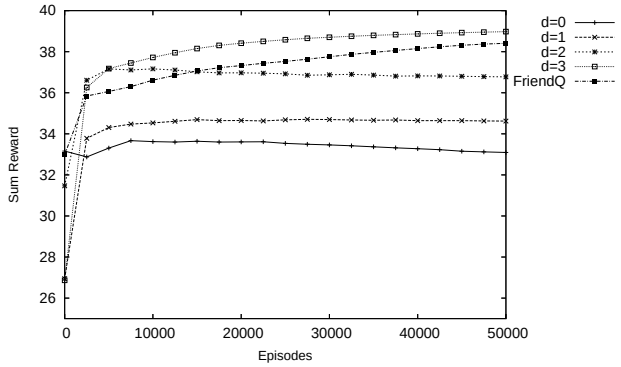

Fig. 3. Rewards for Total Joint Action Q-learning for problem P_1

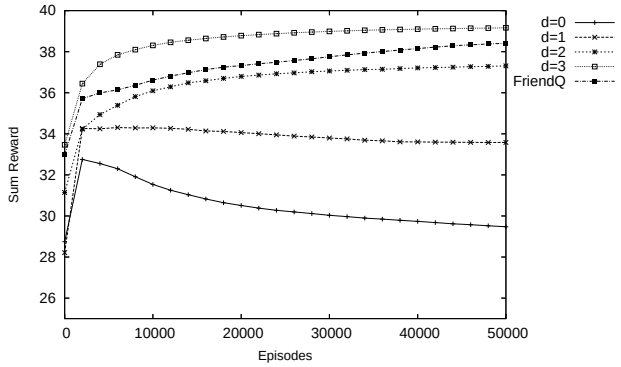

Fig. 4. Rewards for Partial Joint Action Q-learning for problem P_1

Figure 3 shows the result of TJA Q-learning with distance from $d = 0$ to $d = 3$. This algorithm is compared to the total observation problem resolved by Friend Q-Learning. For $d = 0$, $d = 1$ and $d = 2$, TJA converges to a local maximum, which increases with d. In these cases, the approximated values are respectively about 86%, 89% and 94% of the optimal value. When $d = 3$, that is, when the local view is equivalent to the totally observable view, the average sum rewards converges to the total sum rewards of Friend Q-learning. However, since we do not take into account the absolute position of the center agent, TJA converges quickly than Friend Q-learning. Figure 4 shows the results of PJA Q-Learning on the same problem. As previously, for $d = 0$, $d = 1$ and $d = 2$, PJA converges to local maxima respectively about 76%, 86% and 97% of the optimal value. These values are lower than TJA's values but, for $d = 2$, the value is still close to the optimal.

For the second result, we compare PJA Q-learning for two different problems. We define a correct approximation distance d_{app} for each problem, where the associated policy is closed to the optimal value. The first problem is the same as previously (Figure 4) and we can show that $d_{app} = 3$ for this problem. In the second problem P_2, we enlarge the number of lanes and the length of the road ($X = 5, Y = 20, V = 0, \cdots, 5$ and $N = 3$). This problem increases the number of states but decreases the possible interactions between vehicles because they have more space. For the second problem P_2, Figure 5 shows the comparison between Friend Q-learning and PJA Q-learning from $d = 0$ to $d = 7$. We can see that from $d = 4$, there is only small differences between PJA and Friend Q-learning. Consequently, for this problem, we can see that $d_{app} = 4$. The problem of this approach is the need of calculating the optimal policy, which can be intractable, to get d_{app}.

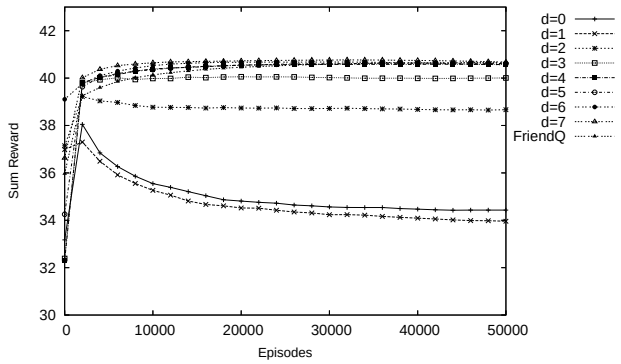

Fig. 5. Rewards for Partial Joint Action Q-learning for problem P_2

As we can see, we need to generalize this result to know the d_{app} parameter without calculating the optimal policy, which can be absolutely intractable for

big problems. To present the third result, we calculate the ratio $DS = XY/N$ which represents the degree of space for each agent. Obviously, if the space (X or Y) increases, then each agent has more space for itself. As we study a problem where the team of agent has to handle only negative interaction, the higher the ratio, the more space agents have. We compare the performance of our PJA algorithm for different ratios. The ratio for the two first problems is respectively $DS_{P_1} = 7$ and $DS_{P_2} = 33$. We add two new problems P_3 ($X = 5$, $Y = 20$, $V = 0, \cdots, 5$ and $N = 5$) and P_4 ($X = 6, Y = 28, V = 0, \cdots, 5$ and $N = 4$) where the ratio are respectively 20 and 42. Table 1 presents the results for each problem after 50000 episodes. For each problem, we define the correct approximation distance d_{app} such as $1-(\frac{R_{d_{app}}}{R_{friendQ}}) < \epsilon$. When $\epsilon = 0.01$, $d_{app}^{P_1} = 3$, $d_{app}^{P_2} = 4$, $d_{app}^{P_3} = 2$ and $d_{app}^{P_4} = 2$.

Algorithms	P_1	ϵ_{P_1}	P_2	ϵ_{P_2}	P_3	ϵ_{P_3}	P_4	ϵ_{P_4}
FriendQ	38.4 ± 1.1	-	40.6 ± 0.3	-	37.0 ± 1.2	-	37.6 ± 0.3	-
PJA $d = 7$	-	-	40.6 ± 0.2	$\sim 0\%$	37.2 ± 0.7	$\sim 0\%$	38.4 ± 0.2	$\sim 0\%$
PJA $d = 6$	-	-	40.5 ± 0.2	$\sim 0\%$	37.9 ± 0.7	$\sim 0\%$	38.8 ± 0.4	$\sim 0\%$
PJA $d = 5$	-	-	40.6 ± 0.2	$\sim 0\%$	37.8 ± 0.9	$\sim 0\%$	38.7 ± 0.4	$\sim 0\%$
PJA $d = 4$	-	-	40.5 ± 0.2	$\sim 0\%$	38.3 ± 0.8	$\sim 0\%$	38.7 ± 0.2	$\sim 0\%$
PJA $d = 3$	39.1 ±0.2	$\sim 0\%$	40.0 ± 0.2	$\sim 2\%$	38.7 ± 0.6	$\sim 0\%$	38.9 ± 0.2	$\sim 0\%$
PJA $d = 2$	37.3 ±0.2	$\sim 3\%$	38.6 ± 0.2	$\sim 5\%$	37.7 ± 0.5	$\sim 0\%$	38.5 ± 0.1	$\sim 0\%$
PJA $d = 1$	33.5 ±0.2	$\sim 14\%$	33.9 ± 0.3	$\sim 15\%$	35.2 ± 0.3	$\sim 5\%$	35.1 ± 0.4	$\sim 8\%$
PJA $d = 0$	29.4 ±0.3	$\sim 24\%$	34.4 ± 0.4	$\sim 15\%$	33.5 ± 0.4	$\sim 10\%$	34.3 ± 0.3	$\sim 11\%$

Table 1. Average Rewards and standard deviation after 50000 episodes

To discover a relation between the ratio DS and the value of d_{app}, we compare in Figure 6, the link between DS and the degree of observability defined as $\frac{d_{app}}{d_{total}}$ where d_{total} is the maximal distance for a given problem. For example, d_{total} for the problem P_1 is 3. We can see that the degree of observability decreases with the degree of space for each agent. We calculate an interpolated curve assuming that the degree of observability cannot be higher than 1 when $DS < 7$. We can see that the needed observability decreases and tends to 0 when DS increases. With this relation between both parameters, observability and degree of space, we can evaluate, for other problems how would be the d_{app} value.

Thus, introducing the locality of the view allows us to limit the observability of the state. More precisely, this approach allows us to use partial version of Friend Q-learning in real world problems where the state is always partially observable. We obtain an approximation of the optimal policy without knowing the transition function. This approximation can be very close to the optimal policy.

In our approach, we do not take into account communication explicitly for many reasons. First of all, in real world problem, choosing the right communication cost is not an easy task. Furthermore, as we said previously, the communica-

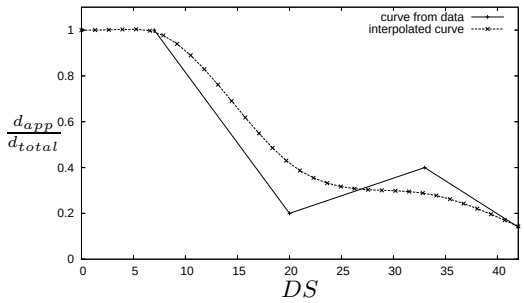

Fig. 6. Link between observability and degree of space

tion cost depends not only on the sent message but also on the distance between sender and receivers. This problem complicates design of communication cost. Consequently, knowing the value of the approximated policy and the associated communication policy (and consequently, the cost of this policy) to obtain the n-partial state, the multiagent system designer can find a good approximation for the real world problem.

6 Related Work

The most general model which is related to our work is Partially Observable Stochastic Games (POSG). This model formalizes theoretically the observations for each agent. The resolution of this kind of games has been studied by Emery-Montermerlo [12]. This resolution is an approximation using Bayesian games. However, this resolution is still based on the model of the environment unlike our approach which do not take into account this information explicitly since we assume that the environment is unknown.

Concerning the space search reduction, Sparse Cooperative Q-Learning [13] allows agents to coordinate their actions only on predefined set of states. In the other states, agents learn without knowing the existence of the other agents. However, the states where the agents have to coordinate themselves are selected statically before the learning process, unlike in our approach. The joint actions set reduction has been studied by Fulda and Ventura who propose Dynamic Joint Action Perception (DJAP) [14]. DJAP allows a multiagent Q-learning algorithm to select dynamically the useful joint actions for each agent during the learning. However, they concentrated only on joint actions and they tested only their approach on problems with few states.

Introducing communication into decision has been studied by Xuan, Lesser, and Zilberstein [1] who proposed a formal extension to Markov Decision Process with communication when each agent observes a part of the environment but all agents observe the entire state. Their approach proposes to alternate communication and action in the decentralized decision process. As the optimal policy

computation is intractable, the authors proposed some heuristics to compute approximation solutions. The main differences with our approach is the implicit communication and the model-free learning in our approach. More generally, Pynadath and Tambe [2] has proposed an extension to distributed POMDP with communication called COM-MTDP, which take into account the cost of communication during the decision process. They presented some complexity results for some classes for team problems. As Xuan, Lesser, and Zilberstein [1], this approach is mainly theoretical and does not present model-free learning.

The locality of interactions in an MDP has been theoretically developed by Dolgov and Durfee [15]. They presented a graphical approach to represent the compact representation of an MDP. However, their approach has been developed to solve an MDP and not to solve directly a multiagent reinforcement learning problem where the transition function is unknown.

Regarding the reinforcement learning in a vehicle coordination problem, Ünsal, Kachroo and Bay [16] have used multiple stochastic learning automata to control longitudinal and lateral path of one vehicle. However, the authors did not extend their approach to multiagent problem. In his work, Pendrith [17] presented a distributed variant of Q-Learning (DQL) applied to lane change advisory system, that is closed to our problem described in this paper. His approach uses a local perspective representation state which represents the relative velocities of the vehicles around. Consequently, this representation state is closely related to our 1-partial state representation. Contrary to our algorithms, DQL does not take into account the actions of the vehicles around and update Q-Values by an average backup value over all agents at each time step. The problem of this algorithm is the lack of learning stability.

7 Conclusion

In this article, we proposed an approach to evaluate a good approximation of a multiagent decision process, introducing a degree of observability for each agents. Without taking into account explicit communication to obtain a degree of observability, we proposed Friend Q-learning algorithms extension which uses only observable state and observable actions from the other agents. We show that only partial view is needed to obtain a good policy approximation for some team problems, especially the changing lane problem between vehicles. We show a relation between a good degree of observability and the space allowed for each agent. However, this relation is only empirical and our approach is only restricted to negative interaction management problems. It is possible that in other problems, this relation could be different.

Adapting multiagent learning algorithm for real world problems is really challenging and many works need to be done to achieve this goal. For future work, we plan to evaluate more theoretically the relation between the degree of observability, the performance of the learned policy and the speed of learning. To define some formal bounds, we will certainly need to use complex communication cost. Finally, introducing the distance for a measure of observability is basic. We

plan to discover others kind of distance between observability to generalize our approach to positive and negative interaction management problems in teams. Also, it will be very interesting to study the effect of partial local view to non cooperative cases.

References

1. Xuan, P., Lesser, V., Zilberstein, S.: Communication decisions in multi-agent cooperation: Model and experiments. In Müller, J.P., Andre, E., Sen, S., Frasson, C., eds.: the Fifth International Conference on Autonomous Agents, Montreal, Canada, ACM Press (2001) 616–623
2. Pynadath, D.V., Tambe, M.: The communicative multiagent team decision problem: Analyzing teamwork theories and models. Journal of AI Research **16** (2002) 389–423
3. Aras, R., Dutech, A., Charpillet, F.: Cooperation in Stochastic Games through Communication. In: fourth Internantional Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05) (poster), Utrecht, Nederlands (2005)
4. Verbeeck, K.: Exploring Selfish Reinforcement Learning in Stochastic Non-Zero Sum Games. PhD thesis, Vrije Universiteit Brussel (2004)
5. Bui, H.H.: An Approach to Coordinating Team of Agents under Incomplete Information. PhD thesis, Curtin University of Technology (1998)
6. Littman, M.: Friend-or-Foe Q-learning in General-Sum Games. In Kaufmann, M., ed.: Eighteenth International Conference on Machine Learning. (2001) 322–328
7. Wang, X., Sandholm, T.W.: Reinforcement Learning to Play An Optimal Nash Equilibrium in Team Markov Games. In: 16th Neural Information Processing Systems: Natural and Synthetic conference. (2002)
8. Moriarty, D.E., Langley, P.: Distributed learning of lane-selection strategies for traffic management. Technical report, Palo Alto, CA. (1998) 98-2.
9. Varaiya, P.: Smart cars on smart roads : Problems of control. IEEE Transactions on Automatic Control **38**(2) (1993) 195–207
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA (1998)
11. Basar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. 2nd edn. Classics In Applied Mathematics (1999)
12. Emery-Montermerlo, R.: Game-theoretic control for robot teams. Technical Report CMU-RI-TR-05-36, Robotics Institute, Carnegie Mellon University (2005)
13. Kok, J.R., Vlassis, N.: Sparse Cooperative Q-learning. In Greiner, R., Schuurmans, D., eds.: Proc. of the 21st Int. Conf. on Machine Learning, Banff, Canada, ACM (2004) 481–488
14. Fulda, N., Ventura, D.: Dynamic Joint Action Perception for Q-Learning Agents. In: 2003 International Conference on Machine Learning and Applications. (2003)
15. Dolgov, D., Durfee, E.H.: Graphical models in local, asymmetric multi-agent Markov decision processes. In: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04). (2004)
16. Ünsal, C., Kachroo, P., Bay, J.S.: Simulation study of multiple intelligent vehicle control using stochastic learning automata. IEEE Transactions on Systems, Man and Cybernetics - Part A : Systems and Humans **29**(1) (1999) 120–128
17. Pendrith, M.D.: Distributed reinforcement learning for a traffic engineering application. In: the Fourth International Conference on Autonomous Agents. (2000) 404 – 411